AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method of managing a relational database comprising:

a. receiving queries in a query language, the queries comprising a plurality of query terms;

[b. representing the queries in accordance with a declarative language paradigm;]

b. interpreting the queries by associating a t least one declarative language function with the query terms;

c. converting the queries represented [in a declarative language paradigm] by the at least one declarative language function to [an imperative language] a plurality of imperative language statements; and

d. executing the imperative language [queries] statements.

2. (Currently amended) The method of claim 1 comprising converting the query language to an intermediate [declarative representative] tree representation corresponding to the at least one declarative language function associated with the plurality of query terms, and thereafter converting the query to at least one data structure that is interpreted by an imperative language interpreter core to perform the queries [representation of the queries and executing the imperative language queries].

3. (Currently amended) The method of claim [1] 2, wherein the declarative language function is identified by a pointer to further code such that the declarative language

function is treated as data within the plurality of imperative language statements [comprising converting the query language to an imperative language representation of the declarative language and executing the imperative language queries].

4. (Currently amended) The method of claim 1 wherein the declarative language function is implemented in a declarative language that is chosen from the group consisting of ML, LISP, and HASKELL.

5. (Currently amended) The method of claim 1 wherein the imperative language statements are implemented in an imperative language that is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.

6. (Currently amended) A method of managing a relational database comprising:

a. receiving queries in a query language;

b. converting the query language to [an imperative language representation] a plurality of imperative language statements that represent [of] a declarative language representation of the queries [in accordance with a declarative language paradigm]; and

c. executing the imperative language [queries] statements.

7. (Currently amended) The method of claim 6 wherein the declarative language representation is chosen from the group consisting of ML, LISP, and HASKELL.

8. (Currently amended) The method of claim 6 wherein the imperative language statements are implemented in an imperative language that is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.

9. (Currently amended) A method of managing a relational database comprising:

a. receiving queries in a query language;

b. representing the queries in accordance with a declarative language paradigm as a plurality of declarative language functions;

c. converting the queries represented in [a] the declarative language paradigm to a data structure that is [effectively interpreted using an imperative language] represented by imperative language statements; and

d. executing the imperative language [queries] statements.

10. (Currently amended) The method of claim 9 further comprising converting the query language to an intermediate declarative [representative] representation, and thereafter converting the query to a data structure representing the intermediate declarative representation in the imperative language statements [efficiently interpreted using an imperative language representation of the queries and executing the imperative language queries] .

11. (Currently amended) The method of claim [9] 10, wherein the declarative language functions are identified by pointers to further code such that the declarative language functions are treated as data within the plurality of imperative language statements [comprising converting the query language to an imperative language representation of the declarative language and executing the imperative language queries].

12. (Currently amended) The method of claim 9 wherein the declarative language functions are implemented in a declarative language that is chosen from the group consisting of ML, LISP, and HASKELL.

13. (Currently amended) The method of claim 9 wherein the imperative language statements are implemented in an imperative language that is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.

14. (Currently amended) A database management system adapted to process queries in a pervasive computing environment, said pervasive computing environment comprising at least one client adapted to interact with a server over connection services, said at least one client controlled and configured to

a. receive the queries in a query language, the queries comprising a plurality of query terms;

[b. represent the queries in accordance with a declarative language paradigm;]

b. interpret the queries by associating at least one declarative language function with the query terms;

c. convert the queries represented [in a declarative language paradigm] by the at least one declarative language function to [an imperative language] a plurality of imperative language statements; and

d. execute the imperative language [queries] statements.

15. (Currently amended) The system of claim 14 comprising converting the query language to an intermediate [declarative representative] tree representation corresponding to the declarative language function, and thereafter converting the query to at least one data structure that is interpreted by an imperative language interpreter core to perform the queries [representation of the queries and executing the imperative language queries].

16. (Currently amended) The system of claim [14] 15, wherein the declarative language function is identified by a pointer to further code such that the declarative language function is treated as data within the plurality of imperative language statements [comprising converting the query language to an imperative language representation of the declarative language and executing the imperative language queries].

17. (Currently amended) The system of claim 14 wherein the declarative language function is implemented in a declarative language that is chosen from the group consisting of ML, LISP, and HASKELL.

18. (Currently amended) The system of claim 14 wherein the imperative language statements are implemented in an imperative language that is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.

19. (Currently amended) A database management system adapted to process queries in a pervasive computing environment, said pervasive computing environment comprising at least one client adapted to interact with a server over connection services, said at least one client controlled and configured to

a. receive queries in a query language;

b. represent the queries in accordance with a declarative language paradigm as a plurality of declarative language functions;

c. convert the queries represented in [a] the declarative language paradigm to a data structure that is [interpreted using an imperative language] represented by imperative language statements; and

d. executing the imperative language [queries] statements.

20. (Currently amended) The system of claim 19 further comprising converting the query language to an intermediate declarative [representative] representation, and thereafter converting the query to [an imperative language] a data structure representing the intermediate declarative representation in the imperative language statements [representation of the queries and executing the imperative language queries].

21. (Currently amended) The system of claim [19] 20, wherein the declarative language functions are identified by pointers to further code such that the declarative language functions are treated as data within the plurality of imperative language statements [comprising converting the query language to an imperative language representation of the declarative language and executing the imperative language queries].

22. (Currently amended) The system of claim 19 wherein the declarative language functions are implemented in a declarative language that is chosen from the group consisting of ML, LISP, and HASKELL.

23. (Original) The system of claim 19 wherein the imperative language statements are implemented in an imperative language that is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.

24. (Currently amended) A program product comprising computer readable program code on one or more media, said program code being capable of controlling and configuring a computer system having one or more computers to perform the process of

a. receiving queries in a query language, the queries comprising a plurality of query terms;

[b. representing the queries in accordance with a declarative language paradigm;]

b. interpreting the queries by associating a t least one declarative language function with the query terms;

c. converting the queries represented [in a declarative language paradigm] by the at least one declarative language function to [an imperative language] a plurality of imperative language statements; and

d. executing the imperative language [queries] statements.

25. (Currently amended) The program product of claim 24 comprising converting the query language to an intermediate [declarative representation thereof] tree representation corresponding to the at least one declarative language function associated with the plurality of query terms, and thereafter converting the query to at least one data structure that is interpreted by an imperative language interpreter core to perform the queries [representation of the queries and executing the imperative language queries].

26. (Currently amended) The program product of claim [24] 25, wherein the declarative language function is identified by a pointer to further code such that the declarative language function is treated as data within the plurality of imperative language statements [comprising converting the query language to an imperative language representation of the declarative language and executing the imperative language queries].

27. (Currently amended) The program product of claim 24 wherein the declarative language function is implemented in a declarative language that is chosen from the group consisting of ML, LISP, and HASKELL.

28. (Currently amended) The program product of claim 24 wherein the imperative language statements are implemented in an imperative language that is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.

29. (Currently amended) A program product comprising computer readable program code on one or more media, said program code being capable of controlling and configuring a computer system having one or more computers to perform the process of

a. receiving queries in a query language;

b. representing the queries in accordance with a declarative language paradigm as a plurality of declarative language functions;

c. converting the queries represented in [a] the declarative language paradigm to a data structure that is [interpreted by an imperative language] represented by imperative language statements; and

d. executing the imperative language [queries] statements.

30. (Currently amended) The program product of claim 29 further comprising converting the query language to an intermediate declarative [representative thereof] representation, and thereafter converting the query to a data structure representing the intermediate declarative representation in the imperative language statements [an imperative language representation of the queries and executing the imperative language queries] .

31. (Currently amended) The program product of claim [29] 30, wherein the declarative language functions are identified by pointers to further code such that the declarative language functions are treated as data within the plurality of imperative language statements [comprising converting the query language to an imperative language representation of the declarative language and executing the imperative language queries].

32. (Currently amended) The program product of claim 29 wherein the declarative language functions are implemented in a declarative language that is chosen from the group consisting of ML, LISP, and HASKELL.

33. (Currently amended) The program product of claim 29 wherein the imperative language statements are implemented in an imperative language that is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.